Electronic Version

Stylesheet Version v1.1.1

Description

Virtual Enterprise Computer

COPYRIGHT STATEMENT

BACKGROUND OF INVENTION

[0008] Almost thirty years ago, the inventor was responsible for

the design of mainframe applications. Serious performance and logical difficulties ensued if a computer application was of sufficient size such that it could not fit into a single computer system, In such situations that continue to today, applications spanning more than one computer system present very significant problems in complexity, performance, data integrity, data pollution and synchronization; applications that are spread across multiple computers present enormous difficulties including complex messaging between computers, replication of data that required virtually impossible data synchronicity to maintain data integrity, application synchronicity, complex communications protocols and so on. If a computer could be invented such that all applications and data storage needs could be handled by a single computer entity, major issues such as data pollution, cost, complexity performance could be significantly reduced including simplifying program design and maintenance. No such alternative existed until recently when the inventor invented the Virtual Enterprise Computer which is a method for multiple computers to operate as a single computer entity.

[0009] Business Enterprises rely on computers with conventional von-Neumann computer architecture (memory, control/

logic unit, arithmetic unit and auxiliary storage). Since no one physical computer can handle all the required processing for a business enterprise, business enterprises use many independent computers to satisfy the totality of their processing and data storage needs and requirements. Since data and information processes is spread across independent, separate computer systems each with their own unique information stores and memory; data is duplicated across these computers without synchronicity. This creates a very complex information and technology infrastructure that is very difficult to manage and maintain and presents great difficulties in maintaining data integrity as data is replicated across multiple computer systems and applications. The enormous management and coordination challenges of a disparate group of independent computers promotes data pollution (stale, unorganized data sets) and creates data synchronicity problems and impedance mismatches (incompatibilities) across the breadth of independent computer applications and information systems.

[0010] Each computer is an independent unit encapsulating its own physical memory, disk and other storage devices; further, the functional logic and address space of each

computer is independent and isolated from each other creating significant challenges in sharing information in a meaningful way. In fact, a significant portion of computer programming and resources are committed to provide a rudimentary capability to communicate between computers so data can be transmitted among multiple computers. Even when computers are linked through various communications media including the Internet and Intranets, they lack a common memory and collective, common understanding to support meaningful sharing of information since they lack a common contextual understanding. This environment is akin to a room full of people attempting to work in concert. Much effort is required to get everyone to understand the tasks at hand and work together. A multiple computer environment is very similar. Consider the enormous savings and efficiency that can be achieved if the room of workers could share a common, seamless brain and memory! This notion is at the core of the Virtual Enterprise Computer; the creation of a common, persistent, enterprise-wide, sharable, single source of all data and information.

[0011] Conventional computer methodologies and technologies can only provide the most basic of communications

among computers. They can pass messages but collective understanding is obscure at best. Using the latest communications technologies provide limited addressing resolution only to the computer, not to a computer's internal data structures; conventional computer addressing of data within a computer is isolated from other computers requiring complex, custom computer programs to share data between computers with a common understanding. Even so, there is no enterprise-wide understanding of the data contained within all the computers. Each computer application is a mechanization of a particular function having its own unique interest, justification, interpretation and meaning for the data it uses without concern for the large context of which the data is part. Further, little or no concern is exists for maintaining synchronicity between multiple copies of the same data.

[0012] Even with the World Wide Web, sophisticated search engines are required to gain knowledge of the data contained within other computers' Web Pages, in which the ad hoc searcher provides the context and understanding implicit in the reason for the data search. Moreover, each computer must present pre-established mechanical means, custom web pages, typically HTML and XML docu-

ments to share data and therefore, must anticipate the needs of data searches. No means exist to do ad hoc searches of an entire computer's memory resources to access to the inherent data, information and knowledge contained in the computer because each computer has its own contextual environment of applications and information schema to give meaning to the data it stores.

[0013] In conventional databases, it is computer applications that give meaning to the relationships between the raw data stored in databases. Ad hoc SQL queries are possible. However, searches are limited to the construct of a particular database. Explicit precognizant knowledge must exist so that these searches have meaning.

[0014] Conventional computer applications consist of several distinct parts: input, data processing and output. During the input phase, computer programs create transient information schema that gives meaning to the raw data they read from relational databases. These structures are not easily available; since the programs are transient so are their information schemas. During the output phase of these programs the information schemas are reduced to raw data contained in relational databases. Therefore, only the results of the processing are saved, not the inter-

relationship of the raw data itself, which remain imbedded and inaccessible in the computer applications.

[0015] The typical business enterprise has hundreds to thousands of disparate computer systems acting as file and application servers using legacy mainframe computers based primarily on IBM MVS mainframes, and UNIX and Microsoft Windows file and application servers. Many computers are required because of enormous data processing, and storage demands. The "point solutions" paradigm that argues for each automation implementation to serve individual isolated needs with little or no regard to integration within a larger logical infrastructure and enterprise-wide information sharing creates a fragmented and inconsistent approach to conservation enterprise knowledge and information. Further, the conventional paradigm for computer application design follows closely the paradigm for computer hardware architecture, i.e., timesharing, client/server, web-based computing. Thus, conventional computer software application architecture closely parallels computer hardware technology.

[0016] As we have previously indicated, conventional application and database architecture creates isolated independent units with their own memory context and data storage.

Since each computer has its own independent memory and application processing, complex messaging and data sharing strategies need to be implemented to permit the sharing of data between computer systems. The problem is so complex that no effective and efficient strategy till now has been developed to provide for the convenient and extensible sharing of data and building enterprise-wide knowledge and information schema, that is, the modeling of the business enterprise in real-time (timely business information).

[0017] Moreover, current computer technologies are used to store data, documents and information in independent database and files systems in which each database and file system is isolated and incompatible with other systems and in which computer applications are unaware of other databases and file systems; applications maintain their is own standards for data integrity and definition even though they often replicate data from other computer applications. Thus, data and information is fragmented, unsynchronized and often stale and is not easily combined to provide an enterprise-wide perspective of a business enterprise's operations and resources as defined in its data and information assets, provide a coherent view

of the business enterprise as a whole; they inherently create enormous data pollution problems.

[0018] The Virtual Enterprise-wide Computer obviates the need for creating "point solution" computer automation and eliminated "Islands of Automation". The single Virtual, Enterprise Computer uses my inventions of a Persistent Enterprise(-wide) Memory (Persistent Enterprise Continuum) and Persistent Enterprise-wide Linear Address Space; the VEC creates a data processing environment greatly simplifying the creation and management of enterprise computer applications creating a seamless fabric of knowledge and information that spans the enterprise; it provides persistent information and knowledge schema conveniently available to knowledge workers and executive management to enhance collaboration and reduce errors by providing timely (real-time) business knowledge and information integrated into enterprise-wide business models.

[PRIOR ART]

[0019] Over the last decade, many techniques have been developed in the attempt to address the issues of disparate, isolated computer systems, applications and information systems. These approaches are piece-meal because they

only treat symptoms and only focus on specific business functions, not the entire business enterprise; that is, some of these approaches automate enterprise functions such as supply chain management, they fail to address the needs of the entire enterprise. Further, they all share a hard-wired approach, implementing a fixed, inflexible solution that cannot address changing business needs. Additionally, the information in these systems is difficult to extract as demonstrated by the need to develop web-based front-ends to extract data for other uses. These automation efforts are only enterprise solutions in the most limited of context and include:

[0020] (a) Enterprise Resource Management,

[0021] (b) Human Resource Management,

[0022] (c) Oracle Financial Applications,

[0023] (d) Web-based Front-ends to Back Office Databases,

[0024] (e) Bus Architectures,

[0025] (f) Business Intelligence,

[0026] (g) Client/Server Compute Application Architectures,

[0027] (h) Timesharing, and

[0028] (i) Mainframe Applications.

[0029] None of these approaches provide enterprise-wide, single views of the business enterprise. They are solutions to address specific functional areas hardwiring data to particular business functions. Finally, they are built on the outdated computer technologies such as relational databases.

[0030] Specific disadvantages of prior art include:

[0031] (a) no enterprise-wide persistent shared memory,

[0032] (b) no single enterprise-wide addressing scheme for data, information and knowledge,

[0033] (c) data must be searched and accessed by value not location or address,

[0034] (d) no persistent enterprise-wide shared common data or information schema,

[0035] (e) no enterprise-wide common framework and architecture,

[0036] (f) no historical records, lack of versions of data and information,

[0037] (g) complex computer interconnections and messaging required to disparate computer systems,

[0038] (h) no single enterprise event/logic controller: real-time monitor,

[0039] (i) information stored in its raw data form typically in relational databases making it difficult to extract knowledge,

[0040] (j) application centric: little or no persistence of the business relationships between data elements,

[0041] (k) large monolithic programs in which processing and information schema (relationships between raw data elements) is inaccessible and inflexible,

[0042] (l) cannot easily maintain complex data and information relationships or complex data types including sophisticated graphical information or Computer Aided technical drawings, and

[0043] (m) cannot maintain persistent complex data and information relationships or complex data types including sophisticated graphical information or Computer Aided technical drawings.

[0044] Relational databases cannot create enterprise-wide information schema without significant data replication, building complex, inefficient table structures and reliance on outer-joins between relational database tables. Technological limits of tables within relational databases prevent creating enterprise-wide views of business enterprises without extraordinary means such as replication servers and data replication. Further, outer-joins, the combination

of data across relational database tables - a necessity for building complex, enterprise-wide information schema - significantly impact performance and efficiency.

[OBJECTIVES AND ADVANTAGES]

[0045] The Virtual Enterprise Computer's advantages and objectives include:

[0046] (a) a method for the storage and processes of data and information not limited to the processing and storage limitations of a single convention computer,

[0047] (b) a single common enterprise-wide address space and memory sufficiently robust has to manage a business enterprise's entire processing and data storage needs within a single context,

[0048] (c) a single data storage mechanism and processing environment simplifying building enterprise solutions and integrating all data and information sources into a common enterprise-wide processing and storage environment,

[0049] (d) an extensible, enterprise-wide environment without practical processing or storage limitations,

[0050] (e) a single source for all intellectual assets including but not limited to data, information, knowledge, know-how and documents eliminating redundancy, duplication and inconsistency,

[0051] (f) a seamless enterprise-wide object-oriented computer application environment,

[0052] (g) data centric solution: relationships between data, information knowledge, business process, business workflows, business metrics, and business rules into coherent business models maintained within the Persistent Enterprise-wide Memory (Persistent Enterprise Memory),

[0053] (h) model of the business enterprise aligning intellectual assets with business processes creating knowledge,

[0054] (i) timely (real-time) business information,

[0055] (j) a seamless enterprise-wide information technology infrastructure to conserve business knowledge,

[0056] (k) a single coherent view of the business enterprise,

[0057] (l) flexibility and extensibility to support Change Management Strategies.

SUMMARY OF INVENTION

[0058] The Virtual Enterprise Computer is a model of a single logical computer running all the programs and managing all the intellectual assets of a business enterprise. It is a model which greatly simplifies the implementation and management of enterprise automation solutions.

[0059] The Virtual Enterprise Computer physical embodiment is a

TCP/IP network of UNIX-based and/or Microsoft Windows-based computers operating over private intranets and/or Internet.

[0060] Each computer in the network runs an object database in which data in each computer is identified by a unique identification code. Each object database maintains its own unique object identifiers.

[0061] The object databases in all the computers are integrated into single a persistent enterprise-wide store through the use of a Persistent Enterprise-wide Addressing Space (PEAS) schema that uniquely identifies the location of all data assets within the VEC. PEAS maintains system tables that map logical identifiers of data assets to physical addresses consisting of a machine code address, TCP/IP address and object identifiers within the object databases. That is, each data asset has a PEAS logical identifier that identifies the location of the computer containing the data and its internal location within the computer. Thus, the object databases are logically connected through PEAS creating a single logical storage device spanning the entire network of computers within the VEC called the VEC Main Memory.

[0062] VEC's Main Memory is the VEC's logical processing envi-

ronment. It is a methodology for addressing and managing the persistent elements stored in VEC consisting of Agents encapsulating data and computer code. Each Agent has a persistent logical address. It provides for the logical consistency and persistence of Agents. VEC's logical processing environment ensure referential integrity among Agents, load Agents for execution and perform associated management a security operations.

[0063] Agents are the common architecture for enterprise data and their associated methods. Agents are created and controlled through Application Program Interfaces based on C++ or Java. They are a concatenation of data and code, methods, specific to the data. The Agents provide many functional purposes from interfaces between disparate systems to the Real-Time Monitor.

[0064] Agents are used to create a Persistent Enterprise-wide Memory (PEM): Persistent Enterprise Continuum that contains the complex relationship of all enterprise data; relationships between data assets create information schema, business processes, rules and metrics, and knowledge. The PEM is the business enterprise's knowledge repository with the unique characteristic of being singular, persistent and data centric.

[0065] Relying on the Persistent Enterprise Address Space with the additional concept of self-aware objects called Agents, creates a Persistent Enterprise-wide Memory (Persistent Enterprise Continuum) for the processing and management of all enterprise intellectual assets and which distinguishes this technology from conventional object databases. Agents, the fundamental architectural element of data and information entities with their methods (programs) are managed and located through unique addresses within the Persistent Enterprise Memory; they interact to accomplish specific processing of intellectual assets; the agents are maintained in the VEC's main memory managed and controlled by the Event/Logic Controller which decides which Agents are to execute their program code based on the occurrence of pre-established events within the business enterprise. The Event/Logic Controller loads Agents to be executed in a wrapper program with the Controller, executing the desired code.

[0066] Modeling of the business enterprise's tangible and intangible resources, assets, processes and workflows is accomplished by direct one-to-one mapping to Agents, which creates the Persistent Enterprise-wide Memory (Persistent Enterprise Continuum).

[0067] The Virtual Enterprise Computer provides the method and infrastructure to build Enterprise Computer Applications and implement Enterprise Application Integration. By virtual of a single common, enterprise-wide memory, and a common, enterprise-wide processing environment, enterprise-wide applications are made practical. It is an effective solution to business initiatives that here-to-for had no practical implementation including

[0068] (a) Seamless Business Enterprise,

[0069] (b) Real-Time Business Enterprise,

[0070] (c) Single View of the Business Enterprise,

[0071] (d) Single View of Customers and Suppliers,

[0072] (e) Enterprise Application Integration,

[0073] (f) Straight-Through-Processing,

[0074] (g) Coherent Business Enterprise,

[0075] (h) Optimized Business Enterprise,

[0076] (i) Intelligent Business Enterprise,

[0077] (j) Enterprise Business Intelligence,

[0078] (k) Mergers and Acquisitions,

[0079] (l) Collaborative Work Environments,

[0080] (m) Enterprise Resource Planning,

[0081] (n) Enterprise Business Modeling,

[0082] (o) Enterprise Knowledge Management, and

[0083] (p) Enterprise Business Intelligence.

Brief Description of Drawings

[0084] Figure 1, Virtual Enterprise Computer Overview, shows the major components of the invention indicating key relationships between the major components.

Detailed Description

[0085] The Virtual Enterprise Computer is a model of a single logical computer running all the programs and managing all the intellectual assets of a business enterprise including its virtual components (partners and vendors). It is a model which greatly simplifying the implementation and management of enterprise automation solutions.

[0086] The Virtual Enterprise Computer physical embodiment is a TCP/IP network of UNIX-based and/or Microsoft Windows-based computers operating over private intranets and/or Internet.

[0087] Each computer in the network runs an object database in

which data in each computer is identified by a unique identification code. Each object database maintains its own unique object identifiers.

[0088] The object databases in all the computers are integrated into single a persistent enterprise-wide store through the use of a Persistent Enterprise-wide Addressing Space (PEAS) schema that uniquely identifies the location of all data assets within the VEC. PEAS maintains system tables that map logical identifiers of data assets to physical addresses consisting of a machine code address, TCP/IP address and object identifiers within the object databases. That is, each data asset has a PEAS logical identifier that identifies the location of the computer containing the data and its internal location within the computer. Thus, the object databases are logically connected through PEAS creating a single logical storage device spanning the entire network of computers within the VEC called the VEC Main Memory.

[0089] VEC's Main Memory is the VEC's logical processing environment. It is a methodology for addressing and managing the persistent elements stored in VEC consisting of Agents encapsulating data and computer code. Each Agent has a persistent logical address. It provides for the

logical consistency and persistence of Agents. VEC's logical processing environment ensure referential integrity among Agents, load Agents for execution and perform associated management a security operations.

[0090] Agents are the common architecture for enterprise data and their associated methods. Agents are created and controlled through Application Program Interfaces based on C++ or Java. They are a concatenation of data and code, methods, specific to the data. The Agents provide many functional purposes from interfaces between disparate systems to the Real-Time Monitor.

[0091] Agents are used to create a Persistent Enterprise-wide Memory (PEM): Persistent Enterprise Continuum that contains the complex relationship of all enterprise data; relationships between data assets create information schema, business processes, rules and metrics, and knowledge. The PEM is the business enterprise's knowledge repository with the unique characteristic of being singular, persistent and data centric.

[0092] Relying on this addressing notion with the additional concept of self-aware objects called Agents, creates a Persistent Enterprise-wide Memory (Persistent Enterprise Continuum) for the processing and management of all enter-

prise intellectual assets and which distinguishes this technology from conventional object databases. Agents, the fundamental architectural element of data and information entities with their methods (programs) are managed and located through unique addresses within the Persistent Enterprise Memory; they interact to accomplish specific processing of intellectual assets; the agents are maintained in the VEC's main memory managed and controlled by the Event/Logic Controller which decides which Agents are to execute their program code based on the occurrence of pre-established events within the business enterprise. The Event/Logic Controller loads Agents to be executed in a wrapper program with the Controller, executing the desired code.

[0093] Modeling of the business enterprise's tangible and intangible resources, assets, processes and workflows is accomplished by direct one-to-one mapping to Agents, which creates the Persistent Enterprise-wide Memory (Persistent Enterprise Continuum).

[COMPONENT (STATIC) VIEW]

[0094] The components of the Virtual Enterprise Computer (VEC) including:

[0095] (a) Persistent Enterprise-wide Linear Address Space,

[0096] (b) VEC Main Memory called the Persistent Enterprise Memory: Persistent Enterprise Continuum,

[0097] (c) TCP/IP Network,

[0098] (d) Event/Logic Controller: Real-Time Monitor,

[0099] (e) Self-Aware Agents (Information and Data encapsulated with computer code called Methods),

[0100] (f) Adapter Agents acting as logical interfaces to interconnect legacy data to Agents within the Persistent Enterprise-wide Memory,

[0101] (g) Object-oriented Database acting as a storage medium within each computer resource, and

[0102] (h) Object-oriented Database acting as a storage medium within each computer resource, and Internet/Intranets.

[0103] An address schema that identifies data and information assets by unique logical identification that is mapped to specific physical locations consisting of a computer addresses (unique internet address) and object id within each computer system.

[0104] Address Tables that map unique information and data resource identification numbers to physical addresses on the network.

[0105] The Persistent Enterprise Address Space creates a persis-

tent linear enterprise-wide address space in which each information and data asset has a unique logical identification that is mapped to a unique physical address on a specific computer resource within the internet/intranet indicated in the above diagram. This capability permits the creation of a coherent persistent enterprise memory. Address tables maintain the mapping of logical address to specific physical address. When a data object is required it is acquired through a read operation that finds the physical objects location by querying the Address tables, reads the data and presents it to the requesting program.

[0106] Each information and data asset is a conventional object-oriented entity containing computer application code and data. The combination creates what is called an Agent. Agents execute according the instructions from the Event/Logic Controller, which controls the overall operation of the Virtual Enterprise Computer. A conventional von Neumann computer architecture is created out of a networked collection of computer resources operating on a TCP/IP network.

[0107] A Systems Library provides for the necessary functions to control the overall operation of the Virtual Enterprise Computer and provide program interfaces to provide

standard system functions including the reading, writing and deletion of data.

[OPERATION PREFERRED EMBODIMENT]

[0108] The Virtual Enterprise Computer logically operates as if it is a single computer entity by integrating multiple, physically dispersed computers into a single functional entity by virtue of the Persistent Enterprise-wide Address Space and a collection of system functions provided by the Event/Logic Controller and support libraries. Each physical computer within the VEC infrastructure is running an object database and responds to requests for data. Client computers outside of infrastructure of VEC make the requests for data. The Client computers or the VEC itself depending upon design and performance considerations for each application can perform execution of Agents.

[0109] VEC's Event/Logic Controller: Real-Time Controller (ELC) operates within VEC's Main Memory and is an Agent itself. Once an event occurs, the ELC calls the appropriate Agent(s) for execution. Since all Agents have there own unique logic address, Agents are easily located and accessed by routine software functions provided by the Systems Library to load and execute Agents.

[0110] When data is required from legacy databases, Adapter

Agents are called which read the appropriate information. Similarly, when data needs to be written to legacy databases, Adapter Agents write the data to appropriate databases.

[0111] The VEC provides for security by providing access lists with password control. Additionally, the VEC supports conventional security measures including firewalls and other standard security controls.

[0112] In the event of a computer failure within the VEC, backup and restore capabilities are provided to restore lost data.

[0113] Developing new applications is as simple as creating a collection of new Agents, not unlike creating functions with the C programming language where there is a "main" function. Within the applications, there is a main Agent that is called first and either directly or indirectly controls the execution of other Agents within the application collection of Agents. However, large collections are strictly discouraged since the notion of large monolithic applications work contrary to the principles of VEC. It is primarily the role of the Event/Logic Control to manage the execution of Agents. Thus, what is created is self-aware objects, a genetic programming model, in which Agents are executed upon a specific event occurring.

[0114] Data is fetched through pre-established program calls to Agents within the library of system functions which gets the data for processing by its unique enterprise address or searches for the data by value. Writing and deletion of data occurs in similar fashion.

[DETAILED DESCRIPTION - ALTERNATE EMBODIMENT]

[0115] The Virtual Enterprise Computer relies on object technology and an object-oriented database. However, similar but less efficient results can be achieved using conventional relational database technology. In this case, each data element is given a unique identification number. An index is provided which identifies and links the unique identification number to its physical location within the data stores. Agents still exist but in a more primitive form in which data and information schema are depopulated into basic data stored in relational databases. Alternatively, an object store can be used to manage agents.